

## FIȘA DISCIPLINEI

### 1. Date despre program

Instituția de învățământ superior	Universitatea „Ștefan cel Mare” Suceava
Facultatea	Facultatea de Inginerie Electrică și Știința Calculatoarelor
Departamentul	Departamentul de Calculatoare
Domeniul de studii	Calculatoare și tehnologia informației
Ciclul de studii	Licență
Programul de studii	Calculatoare

### 2. Date despre disciplină

Denumirea disciplinei	<b>INGINERIA PROGRAMELOR</b>				
Titularul activităților de curs	Prof.dr.ing. Cristina TURCU				
Titularul activităților de seminar	Ș.l.dr.ing. Felicia GÎZĂ				
Anul de studiu	IV	Semestrul	7	Tipul de evaluare	E
Regimul disciplinei	Categorია formativă a disciplinei DF - fundamentală, DD - în domeniu, DS - de specialitate, DC - complementară				DD
	Categorია de opționalitate a disciplinei: DI - impusă, DO - opțională, DF - facultativă				DI

### 3. Timpul total estimat (ore alocate activităților didactice)

I a) Număr de ore pe săptămână	5	Curs	3	Seminar		Laborator	2	Proiect	
I b) Totalul de ore pe semestru din planul de învățământ	70	Curs	42	Seminar		Laborator	28	Proiect	

II Distribuția fondului de timp pe semestru:	ore
II a) Studiul după manual, suport de curs, bibliografie și notițe	28
II b) Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren	28
II c) Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri	21
II d) Tutoriat	-
III Examinări	3
IV Alte activități:	-

Total ore studiu individual II (a+b+c+d)	77
Total ore pe semestru (Ib+II+III+IV)	150
Numărul de credite	6

### 4. Precondiții (acolo unde este cazul)

Curriculum	•
Competențe	•

### 5. Condiții (acolo unde este cazul)

Desfășurare a cursului	• PC, videoproiector și ecran, acces la Internet, G Suite, note de curs, bibliografie recomandată	
Desfășurare aplicații	Seminar	•
	Laborator	• PC, videoproiector și ecran, acces la Internet, G Suite, software specializat, lucrări de laborator, bibliografie recomandată
	Proiect	•

### 6. Competențe specifice acumulate

Competențe profesionale	<ul style="list-style-type: none"> <li>• C2. Proiectarea componentelor hardware, software și de comunicații</li> <li>• C4. Îmbunătățirea performanțelor sistemelor hardware, software și de comunicații</li> <li>• C5. Proiectarea, gestionarea ciclului de viață, integrarea și integritatea sistemelor hardware, software și de comunicații</li> </ul>
Competențe	•

transversale

7. **Obiectivele disciplinei** (reieșind din grila competențelor specifice acumulate)

Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>• Însușirea de către studenți a metodelor de analiză, proiectare, implementare și testare a produselor program de calitate, cu respectarea cerințelor, cumulat cu dezvoltarea experienței de programare.</li> </ul>
-----------------------------------	--

8. **Conținuturi**

Curs	Nr. ore	Metode de predare	Observații
<b>1. Introducere, definiție, istoric, principii de bază, fazele ingineriei programării. Dictionar termeni</b>	3h	expunerea, prelegerea, conversație, studiu de caz, demonstrația	
<b>2. Metodologii de dezvoltare software</b>	3h		
<b>3. Managementul proiectelor software</b>	2h		
<b>4. Ingineria cerințelor</b>	3h		
<b>5. Modelarea sistemelor software</b>			
<b>5.1. Introducere, limbaje de modelare</b>	2h		
<b>5.2. UML. Diagrame structurale. Exemple</b>	4h		
<b>5.3. UML. Diagrame comportamentale. Exemple</b>	4h		
<b>6. Proiectarea sistemelor software</b>	3h		
<b>7. Implementarea sistemelor software</b>			
<b>7.1. Prezentare generală, ghiduri</b>	2h		
<b>7.2. Instrumente software</b>	2h		
<b>7.3. Reutilizare software</b>	2h		
<b>8. Șabloane de proiectare</b>	3h		
<b>9. Verificare și validare. Testare software. Tipuri de testare. Instrumente pentru managementul testării</b>	3h		
<b>10. Calitatea produselor software</b>	3h		
<b>11. Securitatea sistemelor software. Securitatea in ciclul de dezvoltare a unui produs software</b>	3h		
<b>Bibliografie</b>			
<ol style="list-style-type: none"> <li>1. A. Cooper, <i>Proiectarea interfețelor utilizator</i>, Ed. Tehnică, București, 1997</li> <li>2. Fabian C., Mihalca R., Chichernea V., Goron S., Botezatu C., Iacob I., <i>Proiectarea sistemelor informatice. Metode de realizare</i>, Sylvi, Bucuresti 2001</li> <li>3. D. Bocu –<i>Inițiere în ingineria sistemelor soft</i>, Ed. Albastră, Cluj-Napoca, 2001</li> <li>4. O. Gherghieș, A. Apetrei, <i>Ingineria programării</i>, Curs, Iași, 2002</li> <li>5. I. Odagescu, F. Furtuna, <i>Ingineria programării</i>, Ed. ASE, București, 2002</li> <li>6. A. Cernian, A.D. Ioniță, <i>Noțiuni aplicate de inginerie a sistemelor de programe</i>, Ed. Matrixrom, București, 2009</li> <li>7. Rod Stephens, <i>Beginning Software Engineering</i> 1st Edition, 2015</li> <li>8. Max Kanat-Alexander, <i>Understanding Software</i>, 2017</li> <li>9. I. Sommerville, <i>Software Engineering</i>, 10th Edition, Pearson, India, 2017</li> <li>10. M. Richards and N. Ford, <i>Fundamentals of software architecture: An engineering approach</i>. Sebastopol, CA: O'Reilly Media, 2020.</li> <li>11. M. Seemann, <i>Code that fits in your head: Heuristics for software engineering</i>. Boston, MA: Addison Wesley, 2021.</li> <li>12. D. Farley, <i>Modern software engineering: Doing what works to build better software faster</i>. Boston, MA: Addison Wesley, 2022.</li> <li>13. Resurse online: tutoriale, articole de specialitate</li> </ol>			
<b>Bibliografie minimală</b>			
1. I.Sommerville, <i>Software Engineering</i> , 10th Edition, Pearson India, 2017			

2. D. Farley, *Modern software engineering: Doing what works to build better software faster*. Boston, MA: Addison Wesley, 2022.

Aplicații (laborator)	Nr. ore	Metode de predare	Observații
1. Prezentare generală. Stabilire temă și echipe de lucru. Familiarizare instrumente software utile	2	lucrări practice, exerciții, studii de caz, evaluare	
2. Elaborarea specificațiilor cerințelor software	2		
3. Proiectarea aplicației. Realizarea diagramelor UML	2		
4. Proiectarea aplicației. Proiectare baza de date. Proiectarea interfeței utilizator.	2		
5. Planificare implementare. Elaborare plan de testare.	2		
6. Implementare	2		
7. Dezvoltare module aplicație	2		
8. Implementare teste	2		
9. Populare bază de date	2		
10. Dezvoltare module aplicație	2		
11. Testare aplicație	2		
12. Rafinare implementare. Pregătire aplicație pentru beneficiar	2		
13. Testarea aplicației de către altă echipă	2		
14. Prezentare și evaluare aplicație și documentație elaborată	2		

Prezența la activitățile practice de laborator este obligatorie, conform regulamentelor USV în vigoare. Conform aceluiași regulamente, activitatea pe parcurs poate fi echivalată, la cerere, prin proiecte, pregătirea și participarea la concursuri profesionale, cu aprobarea cadrului didactic care conduce lucrările practice și cu condiția prezentării unui referat/ proiect, în care sunt descrise activitățile desfășurate și rezultatele obținute, cu evidențierea elementelor specifice tematicii disciplinei.

#### Bibliografie

1. Resurse online: tutoriale, materiale bibliografice specifice limbajelor de programare utilizate, articole de specialitate
2. I.Sommerville, *Software Engineering*, 10th Edition, Pearson India, 2017
3. A. Cernian, A.D. Ioniță, *Noțiuni aplicative de inginerie a sistemelor de programe*, Ed. Matrixrom, București, 2009
4. O. Gherghieș, A. Apetrei, *Ingineria programării*, Curs, Iași, 2002
5. I. Odagescu, F. Furtuna, *Ingineria programării*, Ed. ASE, București, 2002
6. Fabian C., Mihalca R., Chichernea V., Goron S., Botezatu C., Iacob I., *Proiectarea sistemelor informatice. Metode de realizare*, Sylvi, Bucuresti 2001
7. D. Bocu – *Inițiere în ingineria sistemelor soft*, Ed. Albastră, Cluj-Napoca, 2001
8. Booch G., Raumbaugh J., Jacobson I., *The Unified Modelling Language. User Guide*, Addison/Wesley 1999
9. D. Roman – *Ingineria programării obiectuale*, Ed. Albastră, Cluj-Napoca, 1996

#### Bibliografie minimală

1. I.Sommerville, *Software Engineering*, 10th Edition, Pearson India, 2017
2. Manuale de utilizare specifice instrumentelor folosite

#### 9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

- Conținutul cursului și al laboratorului, prin problematica tratată, pune la dispoziția studentului cunoștințe conforme cu așteptările reprezentanților comunității epistemice și angajatorilor reprezentativi din domeniul calculatoarelor și tehnologiei informației. Cursul respectă recomandările IEEE și ACM pentru studiile de Software Engineering. Tematica abordată se regăsește la universități de renume, cum ar fi, University of Manchester; Truman State University; University of Nevada; Politehnica București; Universitatea Politehnica Timișoara; Universitatea Tehnică din Cluj-Napoca; Universitatea Tehnică „Gheorghe Asachi” din Iași; Universitatea Babeș-Bolyai, Facultatea de Matematică și Informatică; Universitatea “Alexandru Ioan Cuza” din Iași, Facultatea de Informatică. Cursul se actualizează anual și în urma discuțiilor cu principalii angajatori, pentru a fi adaptat cerințelor actuale ale pieței.

#### 10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
----------------	----------------------	--------------------	-------------------------

Curs	Înțelegerea și cunoașterea noțiunilor fundamentale Capacitatea de utilizare adecvată a noțiunilor din domeniul ingineriei software	Evaluarea cunoștințelor teoretice și practice din tematica studiată în timpul semestrului prin test grilă	30
		Evaluarea finală a aplicației practice elaborate.	20
Laborator	Evaluarea capacității de analiză, sinteză și concretizare a cunoștințelor teoretice în identificarea soluțiilor problemei propuse la lucrările de laborator Corectitudinea și calitatea dezvoltării aplicației software și documentației aferente	Evaluare continuă a activităților desfășurate	50

## Standard minim de performanță

## Curs

- înțelegerea și cunoașterea noțiunilor fundamentale și a cel puțin 50% din cantitatea de informație vehiculată la orele de instruire
- capacitatea de a utiliza cunoștințele în rezolvarea unor probleme specifice
- elaborarea a minim 2 diagrame UML utilizând un instrument software specific
- capacitatea de a folosi corect un limbaj științific, de specialitate, adecvat disciplinei

## Laborator

- însușirea elementelor de bază prezentate la laborator;
- capacitatea de a utiliza cunoștințele în rezolvarea unor probleme specifice;
- capacitatea de a folosi terminologia specifică domeniului și de a purta o discuție pe teme de specialitate;
- capacitatea de a lucra cu bibliotecile și framework-urile specifice domeniului, folosind unelte software dedicate.

Data completării	Semnătura titularului de curs	Semnătura cadrului didactic coordonator
<b>23.09.2024</b>		

Data avizării	Semnătura responsabilului de program
<b>24.09.2024</b>	

Data avizării în departament	Semnătura directorului de departament
<b>25.09.2024</b>	

Data aprobării în consiliul facultății	Semnătura decanului
<b>27.09.2024</b>	